The FMS HIM User Guide

William Cooke <William.Cooke@noaa.gov>

Robert Hallberg <Robert.Hallberg@noaa.gov>

Table of Contents

FAQs [HIM_faq.html]

1. Introduction

1.1. What is HIM?

This program (HIM) simulates the ocean by numerically solving the Boussinesq primitive equations in isopycnal vertical coordinates and general orthogonal horizontal coordinates. These equations are horizontally discretized on an Arakawa C-grid. There are a range of options for the physical parameterizations, from those most appropriate to highly idealized models for studies of geophysical fluid dynamics to a rich suite of processes appropriate for realistic ocean simulations. The thermodynamic options range from an adiabatic model with fixed density layers to a model with temperature and salinity as state variables and using a full nonlinear equation of state. The uppermost few layers may be used to describe a bulk mixed layer, including the effects of penetrating shortwave radiation. Either a split- explicit time stepping scheme or a non-split scheme may be used for the dynamics, while the time stepping may be split (and use different numbers of steps to cover the same interval) for the forcing, the thermodynamics, and for the dynamics. Most of the numerics are second order accurate in space. HIM can run with an absurdly thin minimum layer thickness (often an Angstrom is used).

Details of the numerics and physical parameterizations are provided in the appropriate source files. Most of the available options are selected by the settings in HIM_init.h and may be overridden at run time through a parsed input file with the same format, although some (such as the equation of state) are selected by specifying which file to use in the path_names file.

There are a range of closure options available in HIM. Horizontal velocities are subject to a combination of horizontal biharmonic and Laplacian friction (based on a stress tensor formalism) and a vertical Fickian viscosity (perhaps using the kinematic viscosity of water). The horizontal viscosities may be constant, spatially varying or may be dynamically calculated using Smagorinsky's approach. A diapycnal diffusion of density and thermodynamic quantities is also allowed, but not required, as is horizontal diffusion of interface heights (akin to the Gent-McWilliams closure of geopotential coordinate models). The diapycnal mixing may use a fixed diffusivity or it may use the shear Richardson number dependent closure described in Hallberg (MWR, 2000). When there is diapycnal diffusion, it applies to momentum as well. As this is in addition to the vertical viscosity, the vertical Prandtl always exceeds 1.

HIM has a number of noteworthy debugging capabilities. Excessively large velocities are truncated and HIM will stop itself after a number of such instances to keep the model from crashing altogether, and the model state is output with a reported time of 9.9e9. This is useful in diagnosing failures, or (by accepting some truncations) it may be useful for getting the model past the adjustment from an ill-balanced initial condition. In addition, all of the accelerations in the columns with excessively large velocities may be directed to a text file.

While HIM is qualitatively similar to MICOM [http://panoramix.rsmas.miami.edu/micom/], most of the details of the numerics differ.

A deliberately long (but somewhat out of date) technical description [http://www.gfdl.noaa.gov/~rwh/HIM/HIM_Description.pdf] of the dynamical core is also available for the brave. This description of the dynamical core is largely based on the appendix of R. Hallberg's thesis, and an appropriate reference for HIM is: Hallberg, R., 1995: Some Aspects of the Circulation in Ocean Basins with Isopycnals Intersecting the Sloping Boundaries. Ph. D. Thesis, University of Washington. 244 pp.

Many of the more recent algorithmic developments with HIM are documented in the peer reviewed literature.

Surface Bulk Mixed Layer

Hallberg, Robert, The suitability of large-scale ocean models for adapting parameterizations of boundary mixing and a description of a refined bulk mixed layer model, *Proceedings of the 2003 Aha Hulikoa Hawaiian Winter Workshop U. Hawaii*, 187-203, 2003
manuscript [http://www.gfdl.noaa.gov/~rwh/papers/Hallberg_Aha2003.pdf]

Thompson, L., K. A. Kelly, D. Darr, and R. Hallberg Buoyancy and mixed-layer effects on the sea surface height response in an isopycnal model of the North Pacific, *J. Phys. Oceanogr.*, **32**, 3657-3670, 2002 .
this paper [http://www.gfdl.noaa.gov/~rwh/papers/thompson0201.pdf]

Interior Diapycnal Mixing

Hallberg, Robert, Time integration of diapycnal diffusion and Richardson number dependent mixing in isopycnal coordinate ocean models, *Mon. Wea. Rev.*, **128**, 1402-1419, 2000.
paper [http://www.gfdl.noaa.gov/~rwh/papers/Richardson_Mixing.pdf]

Legg, S., Hallberg, R.W., and Girton, J.B., Comparison of entrainment in overflows simulated by z-coordinate, isopycnal and non-hydrostatic models, *Ocean Modelling*, *in press*, 2005.
paper [http://www.gfdl.noaa.gov/~rwh/papers/Legg_et_al_05.pdf]

Papadakis,, M. P., Chassignet, E. P., and Hallberg, R.W., Numerical simulations of the Mediterranean Sea outflow: impact of the entrainment parameterization in an isopycnic coordinate ocean model, *Ocean Modelling*, **5**, 325-356, 2003.
paper [http://www.gfdl.noaa.gov/~rwh/papers/Papadakis_et_al_Overflows.pdf]

Nonlinearities of the Equation of State

Hallberg, Robert, A thermobaric instability of Lagrangian vertical coordinate ocean models, *Ocean Modelling*, **8**, 279-300, 2005.
paper [http://www.gfdl.noaa.gov/~rwh/papers/Thermobaric_Instability.pdf]

Horizontal momentum closures

Griffies, S.M. and Hallberg, R., Biharmonic friction with a Smagorinsky-like viscosity for use in large-scale eddy-permitting ocean models, *Mon. Wea. Rev.*, **128**, 2935-2946, 2000.
paper [http://www.gfdl.noaa.gov/~rwh/papers/smg0002.pdf]

Split time stepping

Hallberg, Robert, Stable split time stepping schemes for large-scale ocean modeling, *J. Comp. Phys.*,**35**, 54-65, 1997.

The purpose of this web page is to provide general information about HIM and particular information for how to download and run the code.

1.2. HIM registration

HIM users can acquire the source code and associated data sets from GForge [http://fms.gfdl.noaa.gov/], and are required to register [https://fms.gfdl.noaa.gov/account/register.php] at the GFDL GForge location [http://fms.gfdl.noaa.gov/]. Therefore, users need to register only once to get both the source code and datasets of HIM. More details can be found in the quickstart_guide.html.

1.3. HIM email list

Email concerning HIM are to be directed to the HIM-email list located at `<oar.gfdl.him@noaa.gov>`. All questions, comments, and suggestions are to be referred to this list. An archive of all emails is maintained at email archive [http://fms.gfdl.noaa.gov/mail/?group_id=26]. Note that by registering at GForge to access the code, you are automatically subscribed to the email list.

1.4. HIM: May 2005

This is the first public release of the F90 HIM code. However, the careful conversion of the C HIM code should ensure that there are relatively few bugs for a wholly new code. The Fortran HIM code has been in scientific use at GFDL for about a year now. In addition, the F90 coding style benefits from our experience with both the C HIM code and MOM4. We believe that this will be a valuable and proficient tool for simulating the ocean.

2. Details of HIM

The ~35 source files contain the following subroutines:

HIM/ocean_core/HIM.F90	*step_HIM* contains the main time stepping loops. One time integration option for the dynamics uses a split explicit time stepping scheme to rapidly step the barotropic pressure and velocity fields. The barotropic velocities are averaged over the baroclinic time step before they are used to advect thickness and determine the baroclinic accelerations. The time-averaged barotropic velocity is interactively adjusted within the continuity solver to ensure the correct evolution of the free surface height. At the end of every time step, the free surface height perturbation is determining by adding up the layer thicknesses; this perturbation is used to drive the free surface heights from the barotropic calculation and from the sum of the layer thicknesses toward each other over subsequent time steps, eliminating any discrepancies that were left after the iterations in the continuity solver. The barotropic and baroclinic velocities are synchronized as part of the vertical viscosity algorithm and be recalculating. the barotropic velocities from the baroclinic velocities each time step. This scheme is described in Hallberg, 1997, J. Comp. Phys. 135, 54-65. The other time integration option uses a non-split time stepping scheme based on the 3-step third order Runge-Kutta scheme described in Matsuno, 1966, J. Met. Soc. Japan, 44, 85-88. For problems with more than a very few layers, this non-split scheme is much less efficient than the split scheme, but because it is much simpler and formally more accurate, it is useful to have to verify that temporal truncation errors are not significant. *initialize_HIM* orchestrates the initialization of a HIM run, coordinating among a variety of options. *register_diags* registers the diagnostic variables that are handled by HIM.F90. *set_restart_fields* registers the fields controlled by HIM.F90 that appear in a restart file. *calculate_surface_state* sets up a structure with the appropriate surface properties to be returned as the output of a HIM time step.

HIM/ ocean_driver/HIM_driver.F90	The driver routine is where HIM starts. Inside of *HIM_main* are the calls that set up the run, step the model, and orchestrate output and normal termination of the run.

Initial Condition, Forcing, and Domain Specification Routines

HIM/ ocean_initialization/HIM_initialization.F90	*HIM_initialization* does just that to all of the fields that are needed to specify the initial conditions of the model. HIM_initialization calls a number of other subroutines in HIM_initialization.F90, each of which initializes a single field (or a few closely related fields) that are indicated by the subroutine name. *Get_HIM_Input* gets 5 controlling inputs from a namelist, setting the directories for I/O and the parameter specification file.
HIM/ ocean_driver/HIM_surface_forcing.F90	*set_forcing* sets the current values of surface forcing fields. *wind_forcing* sets the current surface wind stresses. *buoyancy_forcing* sets the current surface heat, fresh water, buoyancy or other appropriate tracer fluxes. *set_forcing_output* sets up the output of any forcing fields. *average_forcing* accumulates time averages of indicated forcing fields. *register_forcing_restarts* is used to specify the forcing-related fields that are written to and read from the restart file.
HIM/ ocean_infra/HIM_metrics.F90	*set_metrics* calculates the horizontal grid spacings and related metric fields, along with the grid point locations. *initialize_masks* initializes the land masks.

Principal Dynamic Routines

HIM/ ocean_core/HIM_CoriolisAdv.F90	*CorAdCalc* calculates the Coriolis and advective accelerations.
HIM/ ocean_core/HIM_PressureForce.F90	*PressureForce* calculates the pressure acceleration.
HIM/ ocean_core/HIM_CompressComp.F90	*register_compress* is used to specify the reference profile of potential temperature and salinity that is used to compensate for compressibility. *uncompress_e_rho* makes internally consistent changes to profiles of interface height and density to offset compressibility and to minimize the

	non-solenoidal pressure gradient term.
HIM/ocean_core/HIM_continuity.F90, HIM_continuity_mpdata.F90, or HIM_continuity_FCT.F90	*continuity* time steps the layer thicknesses.
HIM/ocean_core/barotropic.F90	*btstep* time steps the linearized barotropic equations for use with the split explicit time stepping scheme. *btcalc* calculates the barotropic velocities from the layer velocities. *barotropic_init* initializes several split-related variables and calculates several static quantities for use by btstep. *register_barotropic_restarts* indicates those time splitting-related fields that are to be in the restart file.
HIM/ocean_param/HIM_hor_visc.F90	*horizontal_viscosity* calculates the convergence of momentum due to Laplacian or biharmonic horizontal viscosity. *set_up_hor_visc* calculates combinations of metric coefficients and other static quantities used in horizontal_viscosity.
HIM/ocean_param/HIM_vertvisc.F90	*vertvisc* changes the velocity due to vertical viscosity, including application of a surface stress and bottom drag. *set_viscous_BBL* determines the bottom boundary layer thickness and viscosity according to a linear or quadratic drag law.
HIM/ocean_param/HIM_thickness_diffuse.F90	*thickness_diffuse* moves fluid adiabatically to horizontally diffuse interface heights.
Thermodynamic Routines	
HIM/ocean_param/HIM_diabatic_driver.F90	*diabatic* orchestrates the calculation of vertical advection and diffusion of momentum and tracers due to diapycnal mixing and mixed layer (or other diabatic) processes. mixedlayer, Calculate_Entrainment, apply_sponge, and any user-specified tracer column physics routines are all called by diabatic.
HIM/ocean_param/HIM_diabatic_entrain.F90	*Calculate_Entrainment* calculates the diapycnal mass fluxes due to interior diapycnal mixing processes, which may include a Richardson number dependent entrainment. *Calculate_Rino_flux* estimates the Richardson number dependent entrainment in the absence of interactions between layers, from which the full interacting entrainment can be found.

	Estimate_u_h estimates what the velocities at thickness points will be after entrainment.
	Determine_Kd Specifies the non-shear-dependent diapycnal diffusivity, including bottom-drag sources or a Bryan-Lewis style depth-dependent specification.
HIM/ocean_param/HIM_mixed_layer.F90	*mixed_layer* implements a bulk mixed layer, including entrainment and detrainment, related advection of dynamically active tracers, and buffer layer splitting. The bulk mixed layer may consist of several layers.
HIM/ocean_tracer/HIM_tracer.F90	*register_tracer* is called to indicate a field that is to be advected by advect_tracer and diffused by tracer_hordiff.
	advect_tracer does along-isopycnal advection of tracer fields.
	tracer_hordiff diffuses tracers along isopycnals.
HIM/ocean_param/HIM_tracer_flow_control.F90	*call_tracer_register* calls user-specified subroutines to register tracers for advection and restarts.
	tracer_flow_control_init calls any user-specified tracer initialization subroutines.
	call_tracer_set_forcing calls user-specified subroutines to set up tracer surface (or other) forcing.
	call_tracer_column_fns calls any user-specified tracer column processes subroutines that have been registered.
HIM/ocean_param/USER_tracer_example.F90	*USER_register_tracer_example* demonstrates the registration of tracers for advection and restarts.
	USER_initialize_tracer initializes tracers if they have not previously been read from a restart file.
	tracer_column_physics applies diapycnal advection and diffusion to tracers, potentially including surface fluxes.
HIM/ocean_param/HIM_sponge.F90	*apply_sponge* damps fields back to reference profiles.
	initialize_sponge stores the damping rates and allocates the memory for the reference profiles.
	set_up_sponge_field registers reference profiles and associates them with the fields to be damped.

In HIM/ocean_eqn_state/ HIM/ ocean_eqn_state/ocean_eqn_of_state.F90, ocean_eqn_of_state_linear.F90, or ocean_eqn_of_state_UNESCO.F90	*calculate_density* calculates a list of densities at given potential temperatures, salinities and pressures. *calculate_density_derivs* calculates a list of the partial derivatives with temperature and salinity at the given potential temperatures, salinities and pressures. *calculate_compress* calculates a list of the compressibilities (partial derivatives of density with pressure) at the given potential temperatures, salinities and pressures. *calculate_2_densities* calculates a list of the densities at two specified reference pressures at the given potential temperatures and salinities.
HIM/ ocean_eqn_state/ocean_fit_compressibility.F90	*fit_compressibility* determines the best fit of compressibility with pressure using a fixed 5-coefficient functional form, based on a provided reference profile of potential temperature and salinity with depth. This fit is used in PressureForce.

Infrastructural Routines

HIM/ ocean_infra/HIM_restart.F90	*save_restart* saves a restart file (or multiple files if they would otherwise be too large). *register_restart_field* is called to specify a field that is to written to and read from the restart file. *restore_state* reads the model state from output or restart files. *query_initialized* indicates whether a specific field or all restart fields have been read from the restart files.
HIM/ ocean_infra/HIM_parser.F90	*HIM_parser* parses a parameter specification file with the same format as HIM_init.h to (possibly) change parameters at run time.
HIM/ ocean_infra/HIM_domains.F90	*pass_var* passes a 2-D or 3-D variable to neighboring processors applies corresponding boundary conditions. *pass_vector* passes a 2-D or 3-D pair of vector components or scalars to neighboring processors. *HIM_domains_init* initializes the computational domain. *chksum* sums the bits in an array and writes out the total.

ocean_infra/HIM_diag_mediator.F90	*axes_info* initiates the output axes and stores groupings of them in the ocean grid. *post_data* uses the time-weighting and time_end from enable_averaging in a call to send_data. *enable_averaging* enables averaging for a time interval. *disable_averaging* disables the accumulation of averages. *query_averaging_enabled* indicates whether averaging is currently enabled.
HIM/ ocean_infra/HIM_io.F90 (Input/Output utility subroutines)	*create_file* creates a new file, set up structures that are needed for subsequent output, and write the coordinates. *reopen_file* reopens an existing file for writing and set up structures that are needed for subsequent output.
Purely Diagnostic Routines	
HIM/ ocean_diagnostics/HIM_diagnostics.F90	*calculate_diagnostic_fields* is used to calculate several diagnostic fields that are not naturally calculated elsewhere. *register_time_deriv* is used to register the information needed for diagnostically calculating a time derivative. *calculate_derivs* calculates any registered time derivatives.
HIM/ ocean_diagnostics/HIM_sum_output.F90	*write_energy* writes the layer energies and masses and other spatially integrated quantities and monitors CPU time use. *depth_list_setup* generates a list of the volumes of fluid below various depths.
HIM/ ocean_diagnostics/PointAccel.F90	*write_u_accel* writes a long list of zonal accelerations and related quantities for one column out to a file. This is typically called for diagnostic purposes from vertvisc when a zonal velocity exceeds the specified threshold. *write_v_accel* writes a long list of meridional accelerations and related quantities for one column out to a file. This is typically called for diagnostic purposes from vertvisc when a meridional velocity exceeds the specified threshold.

`_init.h`	sets various constants and parameters for the simulation. Most of these can be overridden at run time by HIM_input and HIM/ocean_infra/HIM_parser.F90.
`HIM/ocean_infra/HIM_memory.h`	contains a number of macros to enable the use of static or dynamic memory allocation.
`HIM/ocean_infra/HIM_metrics_macros.h`	contains the descriptions for a number of metric terms.
`HIM/ocean_param/HIM_hor_visc.h`	contains the descriptions for a number of metric-related fields that are only used in HIM_hor_visc.F90 to calculate horizontal viscosity.

Most simulations can be set up by modifying only the files HIM_init.h, HIM_initialization.F90, and HIM_surface_forcing.F90. These altered files might reside in an example directory. All of the other (unaltered) source code should probably remain in some central directory.

In addition, the diag_table (HIM_diag_table) will commonly be modified to tailor the output to the needs of the question at hand.

The FMS utility mkmf works with a file called path_names to build an appropriate makefile, and path_names should be edited to reflect the actual location of the desired source code.

2.1. Documentation

In addition to this user guide, documentation for HIM is provided by two postscript documents:

1. HIM: The Hallberg Isopycnal Coordinate Primitive Equation Model [http://www.gfdl.noaa.gov/~rwh/HIM/HIM_Description.pdf] by `<Robert.Hallberg@noaa.gov>`. This is the primary reference for HIM. It contains details about some of the numerical algorithms and diagnostics. All usage of HIM in the literature should refer to this document:

    > Add a reference here
    >
    > NOAA/Geophysical Fluid Dynamics Laboratory
    > Available on-line at http://www.gfdl.noaa.gov/~fms.

2.2. Characteristics

HIM possesses a number of computational, numerical, and physical characteristics that are noteworthy. The following provides an overview of the main characteristics of HIM (please refer to A Technical Guide to HIM [http://www.gfdl.noaa.gov/~rwh/HIM/HIM_Description.pdf] for references).

- HIM uses any generalized orthogonal grid, including all metric terms in the Primitive Equations.
- HIM may be coupled to a bulk surface mixed layer through a variable density buffer layer to avoid some of the difficulties associated with detrainment. There may be an arbitrary number of sublayers within the mixed layer, to facilitate the introduction of horizontal processes or biology in the mixed layer representation.
- HIM uses a novel diapycnal time stepping scheme, which is qualitatively correct everywhere in parameter space with no time step limit. There can be multiple baroclinic dynamics time steps between applications of diapycnal processes and tracer advection.
- Richardson number dependent mixing based on Turner's parameterization is an available option. This dramatically improves the representation of mixing in the gravity currents downstream of sills.

- Vertical viscosity is used to essentially eliminate Montgomery potential gradient errors near the intersection of isopycnal surfaces with topography.
- Numerics handle thin layers sufficiently well that a minimum layer thickness of 1 Angstrom works well.
- The Smagorinsky biharmonic viscosity has been implemented. This is a scale-selective, grid- and flow state-dependent viscosity. Only the time steps must now be adjusted to insure stability as resolutions are changed.
- Potential temperature and salinity are state variables, with the user's choice of a linear or fully nonlinear equation of state.
- Tracers are advected with the monotonic flux-form scheme of Easter (1993), implemented in such a way as to avoid any CFL constraint.
- User-provided tracer packages (taking care of vertical tracer processes, boundary conditions, chemistry, biology, etc.) may be added with minimal (~4-line) changes to the generic HIM code.
- HIM is driven with a flexible range of surface fluxes - whatever is appropriate for a particular simulation.
- HIM uses 3-way time splitting: the barotropic mode, baroclinic dynamics, and thermodynamics can all use different time steps. In some instances, this allows the addition of many additional tracers at very modest computational cost.
- HIM meets the code standards set by the GFDL Flexible Modeling System (FMS [http://www.gfdl.noaa.gov/~fms]). It also utilizes a substantial number of software infrastructure modules shared by other FMS-based models. In particular, all I/O (e.g., restarts, forcing fields, initial fields) is handled via NetCDF [http://www.unidata.ucar.edu/packages/netcdf/].
- 2D (latitudinal/longitudinal) horizontal domain decomposition is used for single or multiple parallel processors. HIM has no memory window or slabs.
- For runs coupled with sea-ice or an atmosphere, HIM provides a driving interface that is identical to MOM4's.

2.3. HIM and FMS

HIM has been coded within GFDL's Flexible Modeling System (FMS [http://www.gfdl.noaa.gov/~fms]). Doing so allows for HIM developers to use numerous FMS infrastructure and superstructure modules that are shared amongst various atmospheric, ocean, sea ice, land, vegetative, etc. models. Common standards and shared software tools facilitate the development of high-end earth system models, which necessarily involves a wide variety of researchers working on different computational platforms. Such standards also foster efficient input from computational scientists and engineers as they can more readily focus on common computational issues.

The following list represents a sample of the FMS `shared` modules used by HIM.

- `time manager` [../src/shared/time_manager/time_manager.html]: keeps model time and sets time dependent flags
- `coupler` [../src/coupler/coupler_main.html]: used to couple HIM to other component models
- `I/O` [../src/shared/mpp/mpp_io.html]: to read and write data in either NetCDF, ASCII, or native formats
- `parallelization tools` [../src/shared/mpp/mpp.html]: for passing messages across parallel processors
- `diagnostic manager` [../src/shared/diag_manager/diag_manager.html]: to register and send fields to be written to a file for later analysis

The FMS infrastructure (the "Lima version") has been released to the public on GForge [http://fms.gfdl.noaa.gov/], with further releases every three-four months.

The Flexible Modeling System (FMS [http://www.gfdl.noaa.gov/~fms]) is free software; you can redistribute it and/or modify it and are expected to follow the terms of the GNU General Public License as published by the

Free Software Foundation; either version 2 of the License, or (at your option) any later version.

FMS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with HIM; if not, write to:

> Free Software Foundation, Inc.
> 59 Temple Place, Suite 330
> Boston, MA 02111-1307
> USA
> or see: http://www.gnu.org/licenses/gpl.html

2.4. Test cases

HIM is distributed with a set of test cases located in `src/HIM_examples/`. These tests are taken from models used at GFDL [http://www.gfdl.noaa.gov] for testing the numerical and computational integrity of the code.

Warning

These experiments are *NOT* sanctioned for their physical relevance. They are instead provided for the user to learn how to run HIM, and to verify the numerical and/or computational integrity of the code. *PLEASE* do not assume that the experiments will run for more than the short time selected in the sample runscripts.

`2gyre`: A two-layer, adiabatic, wind-driven double gyre in a basin with sloping sides. There are no thermodynamics in this case. This model is very small and can be easily run on a single processor workstation. It should provide the user with a basic experience of running HIM.

`DOME`: An entraining gravity current, driven by an inflow at the top of a slope. This is a coarse resolution version of one of the cases described in Legg et al., *Ocean Modelling*, 2005. This case has no surface forcing or surface mixed layer submodel, and only buoyancy as a thermodynamic variable, but it is otherwise a full ocean model.

`benchmark`: An idealized wind- and buoyancy-driven high-resolution model. This has all the physics of a full ocean model, but in a configuration that requires no input files. Versions of this have been used by NOAA for benchmarking computers. To change the resolution, only the number of gridpoints and timestep need to be changed.

`global`: A 1-degree, 48-layer ocean-only prototype for a global ocean-climate model on a tripolar grid. This is driven by the CORE forcing and uses restoring of surface properties to climatological values. *THIS HAS NOT BEEN FULLY TUNED, AND IS NOT DIRECTLY USEFUL FOR CLIMATE STUDIES!*.

`him_sis`: The same model as `global`, but coupled to GFDL's sea ice model (SIS). Running this model is somewhat different from HIM-only runs in that flow control is ceded to the standard GFDL coupler, and the HIM code is not used to specify the forcing. From a user's perspective, running this model is virtually identical to running a MOM4 model coupled to SIS.

3. Source code and data sets

3.1. Obtaining source code and data sets

The FMS [http://www.gfdl.noaa.gov/~fms] development team uses a local implementation of GForge [http://fms.gfdl.noaa.gov] to serve FMS software, located at http://fms.gfdl.noaa.gov. In order to obtain the source code and data sets, you must register [https://fms.gfdl.noaa.gov/account/register.php] as an FMS user on our software server. After submitting the registration form on the software server, you should receive an automatically generated confirmation email within a few minutes. Clicking on the link in the email confirms the creation of your account.

After your account has been created, you should log in [https://fms.gfdl.noaa.gov/account/login.php] and request access to the Flexible Modeling System [http://www.gfdl.noaa.gov] project. Once the FMS project administrator grants you access, you will receive a second email notification. This email requires action on the part of the project administrator and thus may take longer to arrive. The email will contain a software access password along with instructions for obtaining the release package, which are described below.

To check out the release package containing source code, scripts, and documentation via CVS, type the following commands into a shell window. You might wish to first create a directory called `fms` in which to run these commands. You should enter the software access password when prompted by the **cvs login** command. At cvs login, the file `~/.cvspass` is read. If this file does not already exist, an error message may display and the cvs login may fail. In this event, you should first create this file via **touch ~/.cvspass**.

```
cvs -z3 -d:pserver:cvs@fms.gfdl.noaa.gov:/cvsroot/him login
cvs -z3 -d:pserver:cvs@fms.gfdl.noaa.gov:/cvsroot/him co -r lima him
```

This will create a directory called `HIM` in your current working directory containing the release package.

If you prefer not to use CVS, you may download the tar file from https://fms.gfdl.noaa.gov/projects/HIM/. Sample output is also available there for download. See Section 5.1, “Sample model output” for more information on the sample output.

All data sets that are needed to run HIM test cases [#HIM test cases] are available for download from the same place in GForge where users get the source code. Therefore, users need to register only once to get both the source code and datasets of HIM. More details can be found in the quickstart_guide.html.

3.2. Description of the data sets

The topography data set for `HIM_global` and `HIM_SIS` is a coarsened version of that kindly provided by Andrew Coward and David Webb at the Southampton Oceanography Centre [http://www.soc.soton.ac.uk/JRD/OCCAM/welcome.html]. Their topography is a montage of that developed by Smith and Sandwell [http://topex.ucsd.edu/marine_topo/mar_topo.html] (1997) by satellite data in the region of 72°S to 72°N, the NOAA (1988) 5-minute global topography ETOPO5 [http://www.ngdc.noaa.gov/mgg/global/etopo5.HTML], and the International Bathymetric Chart of the Arctic Ocean (IBCAO [http://www.ngdc.noaa.gov/mgg/bathymetry/arctic/arctic.html]). All of the forcing is taken from the "Common Ocean Reference Experiments" (CORE) (Large and Yeager, 2005). Temperature and salinity initial and boundary conditions are provided by the NOAA [http://www.noaa.gov] National Oceanographic Data Center (NODC [http://www.nodc.noaa.gov/]) World Ocean Atlas (WOA [http://www.nodc.noaa.gov/OC5/indprod.html]).

4. Preparing the runscript

4.1. The runscript

A runscript is provided in each test case [#HIM test cases] directory (`scripts/him_`*`test_case`*) for each test case [#HIM test cases]. Details can be found in quickstart_guide.html.

Incorporated in the FMS infrastructure is `MPP` [../src/shared/mpp/mpp.html] (Massively Parallel Processing), which provides a uniform message-passing API interface to the different message-passing libraries. If MPICH is installed, the user can compile the HIM source code with MPI. If the user does not have MPICH or the communications library, the HIM source code can be compiled without MPI by omitting the `CPPFLAGS` value `-Duse_libMPI` in the example runscript.

4.2. The diagnostics table

The diagnostics table allows users to specify the sampling rates and choose the output fields prior to executing the HIM source code. It is included in the `experiment` directory for each test case [#HIM test cases] (`HIM/exp/$test_case/HIM_diag_table`). A portion of a sample HIM diagnostic table is displayed below. Reference diag_table_tk.html [../src/shared/diag_manager/diag_table_tk.html] for detailed information on the diagnostics table.

```
"HIM Experiment"
1 1 1 0 0 0
"prog_%4yr_%3dy",     5,"days",1,"days","Time",365,"days"
"ave_prog_%4yr_%3dy", 5,"days",1,"days","Time",365,"days"
"cont_%4yr_%3dy",     5,"days",1,"days","Time",365,"days"

#This is the field section of the diag_table.

# Prognostic Ocean fields:
#=========================

"ocean_model","u","u","prog_%4yr_%3dy","all",.false.,"none",2
"ocean_model","v","v","prog_%4yr_%3dy","all",.false.,"none",2
"ocean_model","h","h","prog_%4yr_%3dy","all",.false.,"none",1
"ocean_model","e","e","prog_%4yr_%3dy","all",.false.,"none",2
"ocean_model","temp","temp","prog_%4yr_%3dy","all",.false.,"none",2
"ocean_model","salt","salt","prog_%4yr_%3dy","all",.false.,"none",2
```

The diagnostics manager module, `diag_manager_mod` [../src/shared/diag_manager/diag_manager.html], is a set of simple calls for parallel diagnostics on distributed systems. It provides a convenient set of interfaces for writing data to disk, namely in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format. The diagnostics manager is packaged with the HIM source code. The FMS diagnostic manager can handle scalar fields as well as arrays. For more information on the diagnostics manager, reference diag_manager.html [../src/shared/diag_manager/diag_manager.html].

4.3. mppnccombine

Running the HIM source code in a parallel processing environment will produce one output NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] diagnostic file per processor. `mppnccombine` [../postprocessing/mppnccombine.c] joins together an arbitrary number of data files containing chunks of a decomposed domain into a unified NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] file. If the user is running the source code on one processor, the domain is not decomposed and there is only one data file. `mppnccombine` [../postprocessing/mppnccombine.c] will still copy the full contents of the data file, but this is inefficient and `mppnccombine` [../postprocessing/mppnccombine.c] should not be used in this case. Executing `mppnccombine` [../postprocessing/mppnccombine.c] is automated through the runscripts [#runscript]. The data files are NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format for now, but IEEE binary may be supported in the future.

`mppnccombine` [../postprocessing/mppnccombine.c] requires decomposed dimensions in each file to have a `domain_decomposition` attribute. This attribute contains four integer values: starting value of the entire non-decomposed dimension range (usually 1), ending value of the entire non-decomposed dimension range, starting value of the current chunk's dimension range and ending value of the current chunk's dimension range. `mppnccombine` also requires that each file have a `NumFilesInSet` global attribute which contains a single integer value representing the total number of chunks (i.e., files) to combine.

The syntax of `mppnccombine` [../postprocessing/mppnccombine.c] is:

```
mppnccombine [-v] [-a] [-r] output.nc [input ...]
```

Table 1. mppnccombine arguments

-v	print some progress information
-a	append to an existing NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] file
-r	remove the '.####' decomposed files after a successful run

An output file must be specified and it is assumed to be the first filename argument. If the output file already exists, then it will not be modified unless the option is chosen to append to it. If no input files are specified, their names will be based on the name of the output file plus the extensions '.0000', '.0001', etc. If input files are specified, they are assumed to be absolute filenames. A value of 0 is returned if execution is completed successfully and a value of 1 indicates otherwise.

The source of `mppnccombine` [../postprocessing/mppnccombine.c] is packaged with the HIM module in the `postprocessing` directory. `mppnccombine.c` should be compiled on the platform where the user intends to run the FMS HIM source code so the runscript [#runscript] can call it. A C compiler and NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] library are required for compiling `mppnccombine.c`:

```
cc -O -o mppnccombine -I/usr/local/include -L/usr/local/lib mppnccombine.c -lnetc
```

5. Examining the output

5.1. Sample model output

Sample HIM model output data files are available to registered [#source code and data sets] HIM users on GFDL's NOMADS server [http://data1.gfdl.noaa.gov/nomads/forms/him_beta.html]. The output data are organized into directories that bear the same names as the test cases [#HIM test cases]. For example, output for test case `2gyre` can be found in directory `sample_output/2gyre`. Output files are classified into two categories:

- `ascii files`: These are files with the suffix .out and are the description of the setup of the run and verbose comments printed out during the run.
- `NetCDF files`: output of the model, both averaged over specified time intervals and snapshots.

Note that these output files are compressed using **tar**. All `.tar` files should be decompressed for viewing. The decompress command is:

```
tar -xvf filename.tar
```

5.2. Analysis tools

There are several graphical packages available to display the model output. These packages vary widely depending on factors, such as the number of dimensions, the amount and complexity of options available and the output data format. The data will first have to be put into a common format that all the packages can read. FMS requires the data to be stored in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format since it is so widely supported for scientific visualization. The graphical package is also dependent upon the computing environment. For ocean modeling, ncview [http://meteora.ucsd.edu/~pierce/ncview_home_page.html], Ferret [http://ferret.wrc.noaa.gov/Ferret/] and Matlab are most commonly used.